

IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of graphics processing, said method comprising:  
  
determining a non-depth conditional status of a fragment corresponding to a pixel;  
  
determining that a scratchpad contains an entry mapped to the pixel[[]; and],  
  
comparing a [[first]] Z value of the fragment to a value of the entry[[],]; and  
  
based on a result of said comparing, passing the fragment to a pixel pipeline,

wherein determining a non-depth conditional status of a fragment includes determining whether incorporation of a second value of the fragment into the pixel is conditional on a non-depth criterion.

2. (Currently Amended) The method of graphics processing according to claim 1, ~~wherein the first value of the fragment includes a Z value~~ said method comprising replacing a line of entries of the scratchpad according to a predetermined cache replacement policy.

3. (Original) The method of graphics processing according to claim 1, wherein the second value of the fragment includes a color value.

4. (Original) The method of graphics processing according to claim 1, wherein determining a non-depth conditional status of a fragment includes determining a current configuration of a pixel pipeline.

5. (Original) The method of graphics processing according to claim 4, wherein determining a current configuration of a pixel pipeline includes determining a value of at least one state variable.

6. (Original) The method of graphics processing according to claim 1, wherein determining a non-depth conditional status of a fragment includes determining whether a non-depth fragment test is enabled.

7. (Original) The method of graphics processing according to claim 1, wherein determining a non-depth conditional status of a fragment includes determining whether an alpha test is enabled.

8. (Currently Amended) The method of graphics processing according to claim 1, wherein said determining a non-depth conditional status of a fragment occurs before said comparing a  $[[first]]$   $\underline{Z}$  value of the fragment to a value of the entry.

9. (Currently Amended) The method of graphics processing according to claim 1, wherein said determining a non-depth conditional status of a fragment occurs after said comparing a  $[[first]]$   $\underline{Z}$  value of the fragment to a value of the entry.

10. (Currently Amended) The method of graphics processing according to claim 1, wherein comparing a  $[[first]]$   $\underline{Z}$  value of the fragment to a value of the entry includes determining whether a Z value of the fragment is less than the value of the entry.

11. (Currently Amended) The method of graphics processing according to claim 1, further comprising overwriting the value of the entry with the  $[[first]]$   $\underline{Z}$  value of the fragment.

12. (Currently Amended) The method of graphics processing according to claim 1, ~~further comprising passing the fragment to a pixel pipeline~~ wherein said determining that a scratchpad contains an entry mapped to the pixel comprises querying the scratchpad with a location value of the fragment.

13. (Original) The method of graphics processing according to claim 1, wherein determining that a scratchpad contains an entry mapped to the pixel includes determining that the entry is valid.

Al  
Ent  
14. (Original) The method of graphics processing according to claim 1, wherein determining that a scratchpad contains an entry mapped to the pixel includes determining that the scratchpad contains a line of entries, the line being mapped to a block of pixels that includes the pixel.

15. (Original) The method of graphics processing according to claim 14, wherein determining that the scratchpad contains a line of entries includes determining that the line is valid.

16. (Original) The method of graphics processing according to claim 14, further comprising initializing a value of each among the line of entries to the backmost among a set of Z values.

17. (Original) The method of graphics processing according to claim 14, further comprising initializing a value of each among the line of entries to the backmost among a set of representative Z values.

18. (Currently Amended) The method of graphics processing according to claim 1, further comprising comparing the ~~[[first]]~~ Z value of the fragment to a representative Z value corresponding to the fragment.

19. (Original) The method of graphics processing according to claim 18, further comprising overwriting the representative Z value.

20. (Original) The method of graphics processing according to claim 19, wherein determining that a scratchpad contains an entry mapped to the pixel includes determining that the scratchpad contains a line of entries, the line being mapped to a block of pixels that includes the pixel, and

wherein overwriting the representative Z value includes comparing the representative Z value with the backmost Z value of the line.

21. (Currently Amended) The method of graphics processing according to claim 20, wherein said comparing the ~~[[first]]~~ Z value of the fragment to a representative Z value occurs before said determining a non-depth conditional status of a fragment.

22. (Original) The method of graphics processing according to claim 1, further comprising initializing the value of the entry to an initial value.

23. (Original) The method of graphics processing according to claim 22, wherein the initial value is a maximum Z value.

24. (Original) The method of graphics processing according to claim 22, wherein the initial value is the backmost among a set of Z values.

25. (Original) The method of graphics processing according to claim 22, wherein the initial value is the backmost among a set of representative Z values.

26. (Original) The method of graphics processing according to claim 22, wherein the initial value is a representative Z value corresponding to a location to which the entry is mapped.

27. (Currently Amended) A method of graphics processing, said method comprising:

determining a non-depth conditional status of a fragment corresponding to a pixel;

determining whether a scratchpad contains an entry mapped to the pixel[[]; and]],

if the scratchpad contains an entry mapped to the pixel, comparing a [[first]] Z value of the fragment to a value of the entry, and otherwise altering a portion of the scratchpad[[],]; and

based on a result of said comparing, passing the fragment to a pixel pipeline,

wherein determining a non-depth conditional status of a fragment includes determining whether incorporation of a second value of the fragment into the pixel is conditional on a non-depth criterion.

28. (Currently Amended) The method of graphics processing according to claim 27, ~~wherein the first value of the fragment includes a Z value, and~~ wherein the second value of the fragment includes a color value.

29. (Original) The method of graphics processing according to claim 27, wherein determining a non-depth conditional status of a fragment includes determining whether a non-depth fragment test is enabled.

30. (Currently Amended) The method of graphics processing according to claim 27, wherein comparing a [[first]] Z value of the fragment to a value of the entry includes determining whether a Z value of the fragment is less than the value of the entry.

31. (Original) The method of graphics processing according to claim 27, further comprising initializing the value of the entry to the backmost among a set of Z values.

32. (Original) The method of graphics processing according to claim 27, further comprising initializing the value of the entry to the backmost among a set of representative Z values.

33. (Currently Amended) The method of graphics processing according to claim 27, wherein altering a portion of the scratchpad includes storing the [[first]] Z value of the fragment to the entry.

34. (Original) The method of graphics processing according to claim 27, wherein altering a portion of the scratchpad includes mapping a line of the scratchpad to a block of pixels that includes the pixel.

35. (Original) The method of graphics processing according to claim 34, wherein altering a portion of the scratchpad includes initializing a value of each among the line of entries to the backmost among a set of Z values.

36. (Currently Amended) A method of graphics processing, said method comprising:  
determining a non-depth conditional status of a fragment corresponding to a pixel;  
[[and]]

determining that a scratchpad contains an entry mapped to the pixel;

based on a value of the entry, determining an occlusion status of the fragment[[,]]; and

subsequent to said determining an occlusion status, altering a value of the fragment,

wherein determining a non-depth conditional status of a fragment includes determining whether incorporation of a color value of the fragment into the pixel is conditional on a non-depth criterion.

37. (Original) The method of graphics processing according to claim 36, wherein determining a non-depth conditional status of a fragment includes determining whether a non-depth fragment test is enabled.

38. (Currently Amended) A graphics architecture comprising:

~~an early culler~~ a logic circuit configured and arranged to receive a fragment corresponding to a pixel; [[and]]

a scratchpad ~~configured and arranged to store a value of an entry mapped to the pixel;~~ ;  
and

a pixel pipeline,

wherein the logic circuit is configured to determine that the scratchpad contains an entry mapped to the pixel, and

wherein the ~~early culler~~ logic circuit is ~~further configured and arranged~~ to compare a [[first]] Z value of the fragment to [[the]] a value of the entry, and

wherein the logic circuit is configured to pass the fragment to the pixel pipeline based on a result of the comparing, and

wherein the ~~early-culler~~ logic circuit is ~~further~~ configured and ~~arranged~~ to determine whether incorporation of a second value of the fragment into the pixel is conditional on a non-depth criterion.

39. (Currently Amended) The graphics architecture according to claim 38, wherein ~~the early-culler is configured and arranged to compare a Z-value of the fragment to the value of the entry, and~~

~~wherein the early-culler~~ logic circuit is ~~configured and arranged~~ to determine whether incorporation of a color value of the fragment into the pixel is conditional on a non-depth criterion.

40. (Currently Amended) The graphics architecture according to claim 38, wherein ~~the early-culler~~ logic circuit is configured and ~~arranged~~ to determine whether a non-depth fragment test is enabled.

41. (Currently Amended) The graphics architecture according to claim 38, wherein ~~the early-culler~~ logic circuit is configured and ~~arranged~~ to determine whether a Z value of the fragment is less than the value of the entry.

42. (Currently Amended) The graphics architecture according to claim 38, wherein ~~the early-culler~~ logic circuit is ~~further~~ configured and ~~arranged~~ to initialize the value of the entry to the backmost among a set of Z values.

43. (Currently Amended) The graphics architecture according to claim 38, wherein ~~the early-culler~~ logic circuit is ~~further~~ configured and ~~arranged~~ to initialize the value of the entry to the backmost among a set of representative Z values.



ACW

44. (Currently Amended) The graphics architecture according to claim 38, ~~further comprising a pixel pipeline configured and arranged to receive the fragment from the early~~ wherein the logic circuit is configured to query the scratchpad with a location value of the fragment.

---